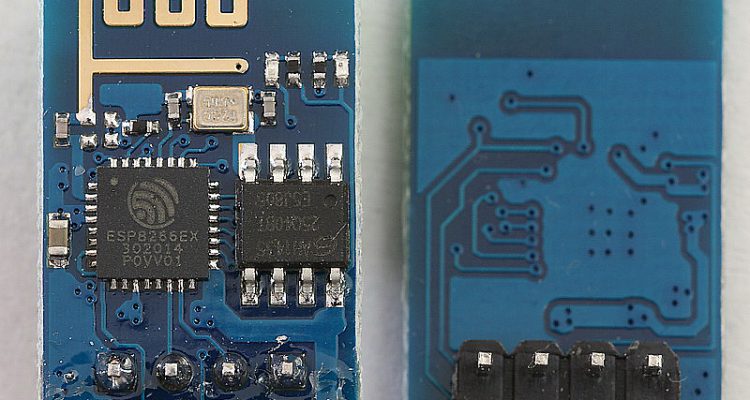
CONTROLLO IOT PER CONDIZIONATORI

<https://elettronicaemaker.it/2017/12/28/controllo-iot-per-condizionatori/>

[E&M-1-2018](https://elettronicaemaker.it/category/em-1-2018/) [Microcontrollori](https://elettronicaemaker.it/category/microcontrollori/) [PIC](https://elettronicaemaker.it/category/pic/)

[0](https://elettronicaemaker.it/2017/12/28/controllo-iot-per-condizionatori/#respond)

[](https://elettronicaemaker.it/wp-content/uploads/2017/12/ESP8266-01.jpg)

di Girolamo D’Orio

*In questo articolo presento un dispositivo IOT universale che permette l’accensione e spegnimento, tramite telecomando, di un condizionatore d’aria.*

Questa realizzazione nasce dall’esigenza di un mio caro amico Geometra, che per esigenze di lavoro si trova spesso all’esterno per fare dei rilievi. Quindi non ha orari precisi di rientro in ufficio, ma gradirebbe trovare un clima confortevole al suo ritorno. Il suo ufficio è composto da almeno tre stanze, delle quali ognuna ha una postazione ben precisa. Un giorno lavora in una, un giorno in un’altra…

Ogni stanza ha un proprio Split, quindi la sua richiesta doveva rispondere a queste particolari esigenze:

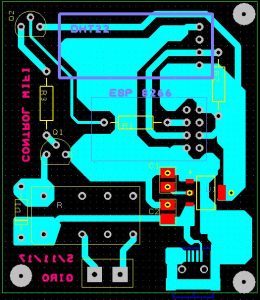
* Voglio spendere poco…
* Con un dispositivo li vorrei pilotare tutti e tre.
* Possibilmente senza passare tanti fili da una parte all’altra dell’ufficio.
* Pratico da utilizzare.
* Deve essere pilotato dal mio Smartphone, ovunque mi trovi.
* Voglio sapere, se possibile la temperatura ambientale in quanto voglio decidere se sia il caso di accendere il condizionatore o meno.

Bene, ho risposto…. Hai un telecomando da sacrificare?

La risposta è stata affermativa in quanto ne aveva due che non funzionavano più, a causa di cadute dalla scrivania. Riparato il telecomando, il quale aveva una saldatura distaccata sulle lamine di appoggio delle batterie stilo, mi sono messo a copiare il codice tramite Arduino. Codice che, tra l’altro, si è rivelato piuttosto complesso rispetto a quello di un tradizionale telecomando tv. La codifica del telecomando dei condizionatori è molto lunga, perché anche se si preme il solo tasto accensione, il dispositivo deve inviare ogni volta tutti i settaggi di impostazioni come temperatura, modalità estate/inverno/velocità della ventola.

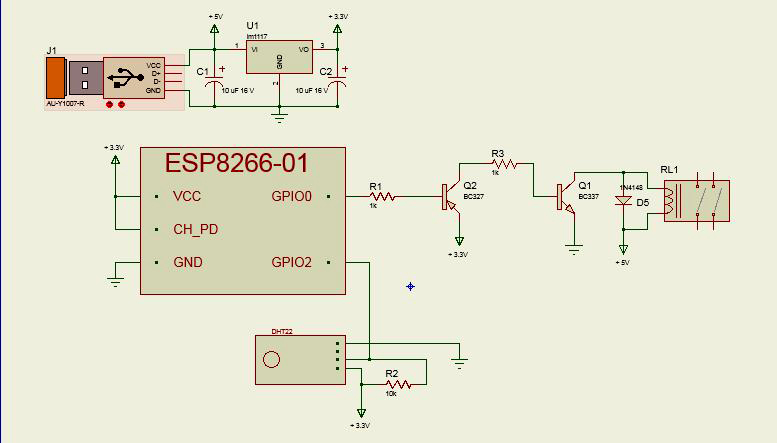
Riflettendo con calma, mi sono chiesto: “ma se ho il telecomando che già fa tutto questo lavoro, perché non lo interfaccio ad un modulo Wi-Fi come l’ESP8266-01?”

Lo rendo alimentabile tramite USB, cosi il mio amico può decidere se alimentarlo con il carica-batterie dello Smartphone o con un Power-Bank. Cosi se lo può spostare da una scrivania all’altra senza creare troppi problemi. Basta indirizzarlo verso il condizionatore ed il gioco è fatto!



Layout del circuito stampato

Schema e descrizione Hardware



Come vedete è veramente semplice, alla portata di tutti. L’alimentazione è fornita tramite USB (5Vcc) quindi per alimentare correttamente Il modulo ESP8266-01 a 3,3V mi sono avvalso del regolatore lineare Lm1117 che essendo un Low Drop-Out lavora bene anche se la sua tensione di ingresso e poco superiore a quella di uscita. Le due capacità da 10µF sono in SMD, purtroppo le ho reperite cosi perché non sono riuscito a trovare di meglio e con poca spesa due elettrolitici a LOW ESR, in quanto sono espressamente richiesti consultando il datasheet del regolatore usato. Nonostante le piccole dimensioni, si saldano tranquillamente bene sul lato piste con un normale saldatore. Il sensore DHT22 con la sua resistenza di pull-up sul pin Data è collegata sul GPiO2 di ESP8266-01 non a caso. Il sensore una volta alimentato ha un invio di livello logico basso di segnale, se collegato al GPIO1 manderebbe il modulo in configurazione di programmazione. Operazione necessaria se si vuole flashare il micro a bordo del modulo, come faremo in seguito.

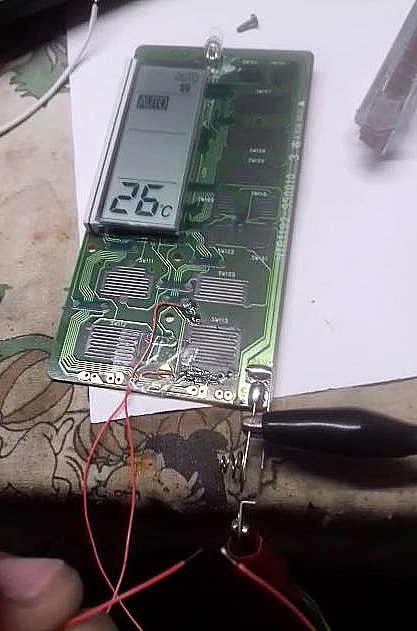
Perché la scelta del relè? Proprio per rendere facile la realizzazione in tutte le occasioni, non sapendo che tipo di telecomando potete avere. Avrei potuto estrapolare come funziona la matrice dei pulsanti che dialoga con il microcontrollore a bordo del telecomando e usare transistor PNP o NPN a seconda del caso, ma avrei reso la vita difficile ad alcuni tra i lettori che intenderanno replicare il progetto. Invece, con il semplice contatto dello scambio NA (normalmente aperto) del relè, sono andato a collegarmi fisicamente sullo stampato del telecomando, in parallelo al pulsante di accensione. Vi chiederete perché due transistor, un PNP che comanda un NPN per poi comandare il relè? Per il solito discorso sopracitato, utilizzando un transistor NPN ESP8266 collegato tramite la resistenza di base al GPI0 il modulo ESP8266 non mi funzionava correttamente. Ho pensato quindi che ritornasse un valore logico Basso che mi mandava il modulo in programmazione. Utilizzando questa configurazione, invece, funziona perfettamente.

|  |
| --- |
| **Elenco Moduli e Sensori** |
| Modulo WIFI ESP8266-01 |
| DHT22 |
| Presa Micro USB tipo B |

|  |  |
| --- | --- |
| **Elenco componenti** | |
| R1 | 10 K Ω ¼ W |
| R2-R3 | 1 KΩ ¼ W |
| C1,C2 | 10 µF 16V elettrolitico |
| U1 | Lm1117 |
| Q1 | BC337 NPN |
| Q2 | BC327 PNP |
| D1 | 1N4148 |
| RL | Relè 5vcc |

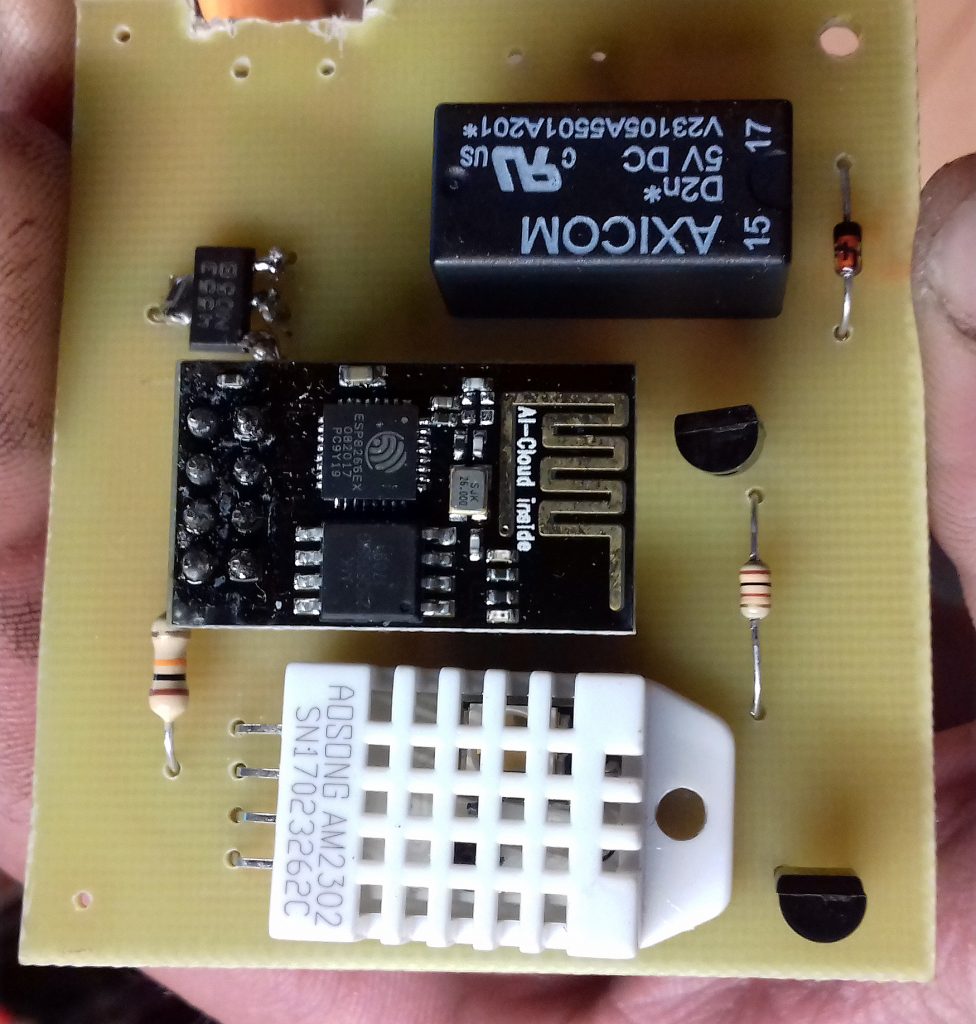
Consigli per il montaggio

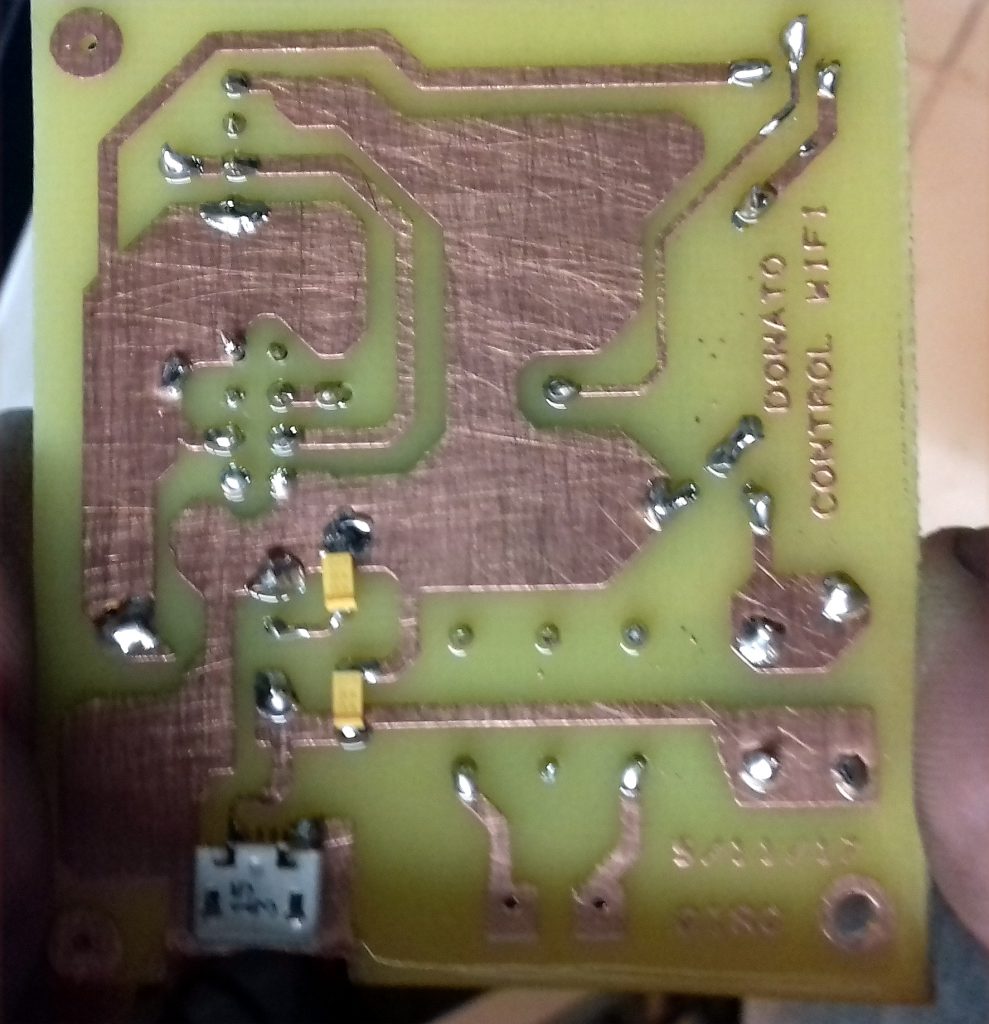
Per prima cosa occorre operare sul telecomando, prestando molta attenzione a non danneggiarlo. Se le piste disegnate del “contatto” del pulsante ON/OFF del telecomando sono in grafite come nel mio caso attenzione, non ci potete saldare direttamente, anzi rischiate di danneggiarle irreparabilmente. Il consiglio che vi propongo è di individuare le piste coperte dal solder mask in un posto dove potete saldare agilmente. Una volta individuate, per rimuovere l’isolamento praticate una leggera abrasione, anche con l’uso di un piccolo cacciavite a taglio, finché il rame sottostante non sarà ben visibile. A quel punto potrete saldarci sopra. Ecco una foto del particolare :



Ricordatevi di saldare le due capacità da 10 µF sul lato piste, prestando attenzione alla giusta polarità, dato che sono elettrolitici. Anche la presa USB è a saldatura superficiale quindi, per evitare un circuito a doppia faccia, occorre saldarla sotto. Per quanto riguarda il regolatore Lm1117 in package SOT23 (e non in TO220), mi sono dovuto arrangiare per saldarlo sopra per permettere una migliore dissipazione. Se lo acquisterete in contenitore TO-220 non avrete problemi, tanto il pin-out è il medesimo.

Ecco come il circuito è completamente montato:





Ecco il dispositivo  fissato sul telecomando:



Descrizione del sorgente

Il sorgente da caricare nel modulo ESP8266 è stato compilato con l’IDE di Arduino. Dato che preparare ESP8266 e l’IDE di Arduino non è una cosa spiegabile in 5 minuti, per non dilungarmi troppo in questo articolo, cercate una guida sul web a vostro piacere o leggete una mia guida al riguardo, in questo sito.

Eseguita alla lettera la precedente guida, ricordatevi su IDE di cambiare il tipo di scheda scegliendo appunto Generic ESP8266 Module, altrimenti il compilatore non potrà compilare neanche  il semplice sorgente che segue, tantomeno caricarlo. Soprattutto perché i fuse ecc. sono totalmente diversi dalla comune scheda di Arduino UNO.

Il sorgente è veramente semplice, grazie all’uso delle due librerie: Una per il sensore DHT22 di temperatura e umidità e l’altra per il modulo ESP8266.

*#include <ESP8266WiFi.h>*

*#include <DHT.h>*

*#define DHTTYPE DHT22*

*#define DHTPIN 2*

*DHT dht(DHTPIN, DHTTYPE, 30);*

*const char\* ssid = “—XXXXXXXXX—-“;//inserire qui la user della vostra wifi*

*const char\* password = “—XXXXXXXXX—-“;//inserire qui la user della vostra wifi*

*int ledPin = 0; // GPIO 0//pin che pilota i transistor—->relè*

*WiFiServer server(80);*

*int stato;*

*int tempmax=21;*

*int tempmin=18;*

*int value = LOW;*

*void setup() {*

*// Serial.begin(115200);*

*pinMode(ledPin, OUTPUT);*

*digitalWrite(ledPin, HIGH);*

*delay(10);*

*dht.begin();*

*delay(10);*

*WiFi.begin(ssid, password);*

*while (WiFi.status() != WL\_CONNECTED) {*

*delay(500);*

*}*

*server.begin();*

*}*

*void loop() {*

*WiFiClient client = server.available();*

*if (!client) {*

*return;*

*}*

*Serial.println(“new client”);*

*while(!client.available()){*

*delay(1);*

*}*

*String request = client.readStringUntil(‘\r’);*

*Serial.println(request);*

*client.flush();*

*delay(4000);*

*float h = dht.readHumidity();*

*float t = dht.readTemperature();*

*int value = LOW;*

*if (request.indexOf(“/LED=ON”) != -1) {*

*digitalWrite(ledPin, LOW);*

*delay(700);*

*digitalWrite(ledPin, HIGH);*

*}*

*client.println(“HTTP/1.1 200 OK”);*

*client.println(“Content-Type: text/html”);*

*client.println(“”);*

*client.println(“<!DOCTYPE HTML>”);*

*client.println(“<html>”);//inizio a scrivere pagina html*

*client.println(“<head>”); //*

*client.println(“<title>GESTIONE CONDIZIONATORE by GIRO</title>”); //*

*client.println(“<head>”); //*

*client.println(“<body>”); //lo sfondo*

*client.println(“<body bgcolor=””#CCFFFF””>”);//colore sfondo*

*client.println(“<center>”);*

*client.println(“<font size=””+4″”>”); client.println(“”); // do not forget this one*

*client.println(“”); // do not forget this one*

*client.println(“”); // do not forget this one*

*client.println(“<br><br>”);*

*client.println(“<font color=””#000000″”>”);*

*client.println(“Cliccare <a href=\”/LED=ON\”> ON </a>  per accendere condizionatore <br>”);*

*client.println(“<center>”);*

*if (isnan(h) || isnan(t) ) {*

*client.println(“<center>”);*

*client.println(“ERRORE sensore,<a href=\”/\”> riaggiornare la pagina </a>”);*

*return;*

*client.println(“<center>”);*

*}*

*client.println(“<center>”);*

*client.print(“Temperatura: “);*

*client.print(t);*

*client.println(” C “);*

*client.println(“<center>”);*

*client.println(“”); // do not forget this one*

*client.print(“Umidita: “);*

*client.print(h);*

*client.println(” %\t”);*

*client.println(“<center>”);*

*client.println(“…<a href=\”/\”>aggiorna DATI </a>…”);*

*client.println(“</font>”);*

*client.println(“<center>”);*

*client.println(“<body>”);*

*client.println(“</html>”);*

*delay(100);*

*}*

Come avrete notato dal sorgente, una volta richiamato l’ESP8266 digitando  l’indirizzo IP assegnato dal Router, il modulo risponde scrivendo una pagina HTML, proprio come se noi la scrivessimo in linguaggio HTML. Tale pagina darà la possibilità di cliccare su ON per accendere il condizionatore e vi comunicherà, in tempo reale, la temperatura e l’umidità misurate dal sensore DHT22. Inoltre, cliccando su aggiorna DATI la pagina subisce un “refresh”, aggiornandosi con le nuove misure di temperatura e umidità.



Non ho aggiunto nessun controllo di stato del dispositivo in quanto, collegandomi solo in parallelo al pulsante, non posso sapere se il condizionatore è acceso. Se il telecomando viene utilizzato manualmente e supponiamo che sia stato acceso, con questi due collegamenti non posso sapere in che stato si può trovare il condizionatore. Anche “memorizzando” internamente tramite software i cambi di stato eseguiti da ESP8266, non posso capire se qualcuno ha premuto manualmente il tasto di accensione o spegnimento. Qui viene però in soccorso il sensore di temperatura e umidità. È piuttosto facile da intuire, poiché se vengono rilevate temperature molto vicine a quelle impostate il condizionatore è acceso, in caso contrario sarà sicuramente spento.

Dato che molti utenti non hanno un indirizzo statico, ma soltanto dinamico, come possiamo richiamare il nostro ESP8266 quando siamo fuori senza grosse difficoltà?

Il mio amico Geometra sopracitato tiene sempre un PC acceso e usa Team Viewer come applicazione per il controllo remoto. E’ un applicazione gratuita, se non viene usata per scopi professionali, mentre il mio amico, essendo un professionista, l’ha acquistata. E’ installabile anche su Smartphone, è facile ed intuibile da utilizzare, non avrete problemi ad usarla. In pratica questa applicazione fa da “ponte” tra il vostro Smartphone e il vostro PC. Tale Software vi permette di gestire, con un controllo totale, il vostro PC e di trasferire eventuali file. Il Programma vi fornisce un ID da richiamare e potete scegliere se utilizzare una password rolling-code o fissa a vostra scelta. La rolling-code e la più sicura, ma più scomoda, dato che ad ogni avvio del programma cambia valore, con il rischio di dimenticarla. Inserendo il programma in esecuzione automatica, per non dimenticarsi di aprirlo tutte le volte, basta un breve blackout di corrente elettrica e il PC si riavvia, ma la password rolling-code sarà diversa e non potrete più accedere. Il mio consiglio è di usare una password fissa a vostra scelta, ma la decisione finale è soltanto vostra

Ecco dove potete scaricare Team Viewer  https://www.teamviewer.com

Una volta entrati nel PC basta semplicemente aprire un qualsiasi browser per la navigazione in Internet e digitare nella barra degli indirizzi l’indirizzo IP che ha assegnato il router al modulo ESP8266.

Se siete collegati, invece con lo Smartphone alla rete Wifi in cui è connesso Esp8266 utillizzate il browser sul vostro cellulare direttamente senza passare da Team Viewer.

Qui troverete lo Sketch da caricare su Esp8266, lo schema disegnato su Proteus 8.0 e il pcb disegnato su DesignSpark 7.2

[](https://elettronicaemaker.it/www.elettronicaemaker.it/wp-content/uploads/SKETCH-SCHEMI-MASTER.rar)

Buon divertimento e buona realizzazione a tutti.